Dimensionality Reduction Using PCA and SVD in Big Data: A Comparative Case Study

Sudeep Tanwar[1(✉)], Tilak Ramani[1], and Sudhanshu Tyagi[2]

[1] Department of CE, Institute of Technology, Nirma University, Ahmedabad, India
{sudeep.tanwar,16mcei19}@nirmauni.ac.in
[2] Department of ECE, Thapar University, Patiala, Punjab, India
s.tyagi@thapar.edu.in

Abstract. With the advancement in technology, data produced from different sources such as Internet, health care, financial companies, social media, etc. are increases continuously at a rapid rate. Potential growth of this data in terms of volume, variety and velocity coined a new emerging area of research, Big Data (BD). Continuous storage, processing, monitoring (if required), real time analysis are few current challenges of BD. However, these challenges becomes more critical when data can be uncertain, inconsistent and redundant. Hence, to reduce the overall processing time dimensionality reduction (DR) is one of the efficient techniques. Therefore, keeping in view of the above, in this paper, we have used principle component analysis (PCA) and singular value decomposition (SVD) techniques to perform DR over BD. We have compared the performance of both techniques in terms of accuracy and mean square error (MSR). Comparative results shows that for numerical reasons SVD is preferred PCA. Whereas, using PCA to train the data in dimension reduction for an image gives good classification output.

Keywords: Dimensionality reduction · Principle component analysis Singular value decomposition · Big data

1 Introduction

Volume of data is increasing exponentially to Tera byte or Peta byte from many sources like biomedicine, social media, Internet of Things (IoT), etc. All data on the planet is growing 40% a year. International data corporation (IDC) has predicted that volume of data will grow above 40 ZB by 2020 [1]. The comparative growth of digital data over time (measured in years) is shown in Fig. 1(a) which is indicates, In 2013 digital universe had 5500 EB, but in 2020 it will be 44 ZB, a 10-fold increment in very short span of time. The top three sources of data are sales & financial transactions (56%), leads & sales contacts from customer databases (51%), and email & productivity applications (tied at 39%). Almost a quarter of respondents (19%) are managing less than a tera byte of data, while only 7% are managing more than a peta byte. Although the average company

Z. Patel and S. Gupta (Eds.): ICFITT 2017, LNICST 220, pp. 116–125, 2018.
https://doi.org/10.1007/978-3-319-73712-6_12

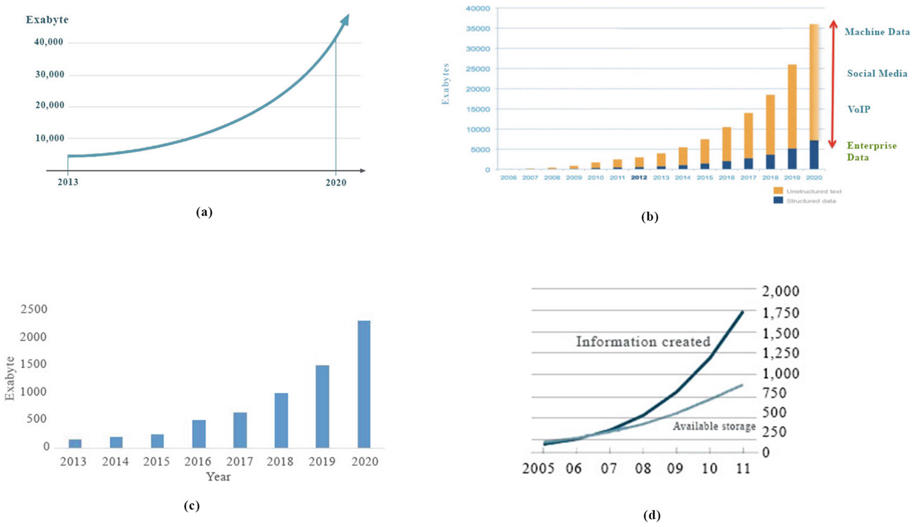

Fig. 1. (a) World wide growth of digital data, (b) Data growth in Enterprise, (c) Data & storage increase over the years, (d) Growth of data in heath care sector

manages 162.9 TB of data, the average enterprise has 347.56 TB of data [1], which is increasing by 33% a year. Figure 1(b) shows the incremental growth in structured and unstructured data over the years. Health care data covers large segment of entire digital universe, and it is increasing 48% in a year. All data in the health care was 153 EB in 2013, but it is expected to be 2,314 EB in 2020. As data volume is growing exponentially, available storage to accommodate it also need to be updated accordingly. Comparative study of growth in data and corresponding storage is shown in Fig. 1(d), indicating that storage is not increasing as rapidly as data. This exponentially increment in data is very complex in several situations like to maintain (a) the real time monitoring for (i) health sector (ii) car parking system (iii) fire alarms, (b) security of (i) offices, (ii) hospitals (iii) defense area and many more. Currently available computing infrastructure and analytical algorithms are not able to manage and process the current form of generated BD. In some situations this data is redundant too, therefore, cleaning of data is required to maintain high quality. Compared to raw data, this cleaned data is very small in size but has important information. To clean this raw data, we have used DR techniques in this paper.

DR is the procedure to convert a dataset have vast number of dimensions into a data subset with less dimensions ensuring no lose of important information. The importance of DR is to improve the accuracy of prediction of classifier, and to decrease the cost of computation. These techniques are basically used to solve machine learning problems to get quality features in classification and regression. Some advantages of DR are summarized as under:

- It compresses data and reduces the storage requirements.
- It reduce the computation time.

- It considers multi-collinearity that gives better performance of the model.
- It eliminates redundant features.
- It helps in eliminating the noise.

To understand the concept of DR we have selected PCA and SVD two different techniques. These techniques are investigated thoroughly, and compared by executing with machine learning algorithm. PCA takes a dataset comprising of the set of tuples focusing on points lying on a high-dimensional space. PCA also searches for the directions with which the tuples line up best. Main objectives of PCA are:

- Form a data table it extricate the vital information.
- Keeping the vital information only, it compress the size of the dataset.
- To simplify description of the dataset.
- To analyze the structure, and factors.

The idea behind PCA is to consider a matrix M be the set of tuples and search for the eigen vectors of MM^T or M^TM. The axis related to the first eigen vector, the one along with which the variance of raw data is maximized. Now, one can apply this transformation to that data. Similarly, the axis related to the second eigen vector is the axis along with which the variance of distances from the first axis is most prominent, and so on. Hence, one can say that PCA is a data mining process. The high dimensional data is supplanted by the projection on essential axes. These axes are related to the largest eigenvalues. Finally, raw data is estimated by data that has less dimensions compared to raw data.

On the other side, SVD is a method to distinguish the dimensions along with which data points show the highest variation. SVD permits to get the best estimation of the raw data using less dimensions. This approach permits a correct portrayal of any matrix. Furthermore, this approach removes the less essential dimensions of that portrayal to create an approximate portrayal with any coveted dimensions. SVD decompose an $m \times n$ matrix, M into U, S, and V. This decomposition has the form USV^*. Here, U is an $m \times r$ matrix, S is a $r \times r$ diagonal matrix, & V is an $n \times r$ matrix. We can utilize them to diminish the number of vectors to the variance we actually required. Diminishing the number of vectors can remove noise from the raw dataset.

1.1 Research Contribution of Paper

Contributions of this paper are as follows:

- We have reduced the dimension of sparse and dense dataset using PCA and SVD.
- We have compared the performance of PCA and SVD by applying them on two different dataset.

The rest of the paper is structured as follows. Section 2 highlights previous work done by researchers in this domain with pros and cons of individual. Section 3 highlights the need of DR and present the techniques PCA and SVD. Section 4 presents the comparison result of both techniques in terms of accuracy & mean square error and finally Sect. 5 concluded the paper.

Table 1. Comparison of existing approaches

Author	Problem statement	Solution	Drawback
Swati *et al.* [2]	The classification of high dimensional data give wrong outcomes	A method that utilizes DR techniques	Another classifier for classification can be used instead of ARTMAP to reduce more time
Person *et al.* [4]	Show points in plane or higher dimensional space by the straight line or plane	Principle component analysis	It becomes more cumbersome when we have more variables which involves the determination of least root
Oja *et al.* [7]	PCA for neural networks	A completely parallel (nonhierarchical) design that gets orthogonal vectors spanning an m-dimensional PCA subspace	Lateral connections between the units are not considered
Sanger *et al.* [8]	Measure the data in network results can be troublesome without exact learning of the distribution on the input data	Optimality principle for training an unsupervised feedforward neural network	The algorithm is only for single-layer linear networks
Henry *et al.* [13]	Identify the dimensions along which data points	Singular value decomposition	When there is no change in one of the axes, SVD fails
Deerwester *et al.* [14]	Dimensionality reduction issue with regards to information retrieval	Use SVD for making features representing multiple words and after that comparing them	Implementation issues will emerge as in raw vector methods, the estimation of such retrieval improving methods must be reevaluated
Sarwar *et al.* [17], Brand *et al.* [16]	The high cost of finding the SVD	Update an existing SVD without recomputing it from scratch	Works well for some recommender applications and less well for others

2 Related Work

This section highlights the work done by various researchers in this domain. Swati *et al.* [2] classified the high dimensional raw data that creates incorrect outcomes. To obtain precise outcomes, high dimensional raw dataset should be compressed to enhance the accuracy of outcome. Repetitive and the conflicting data should be eliminated to achieve it. In [2] authors have presented a constraint selection algorithm to utilizing DR techniques. Because of the DR techniques the computation time is reduced. Tarun *et al.* [3] took the DR technique diminish space and improves the overall performance. For DR meta-heuristics techniques were utilized. To reduce the space DR technique is more valuable; fast information retrieval, optimized image processing, good visualization, exact classification for area oriented datasets. PCA for DR was introduced by Pearson *et al.* [4] and modern representation was given by Hotelling *et al.* [5]. Selection of the dimensions using PCA was explained by Jolliffe *et al.* [6]. One dimensional PCA was implemented for neural networks by Hebb learning *et al.* [7] and later on extended to hierarchical multidimensional PCA by Sanger [8–10]. Further, in [7] authors have given a completely parallel plan that concentrates on orthogonal vectors traversing an m-dimensional PCA subspace. Baldi *et al.* [11] demonstrated the error surface for linear, three layer auto-associators with hidden layers of width m has global minima relating to input weights that traverse the m-dimensional PCA subspace.

SVD was first introduced by Golub *et al.* [12] and later on Henry & Hofrichter [13] utilized it to recognized the dimensions along which data points shows the largest variation. Deerwester *et al.* [14] analyzed the DR issues with regards to information retrieval. They were compared documents using the words they consist of, and they proposed a method of producing features representing different words and then comparing them. Recently, Sarwar *et al.* [15] used SVD for recommender systems. One of the difficulties of utilizing an SVD-based algorithm for recommender systems is the high cost to search the SVD. In spite of the fact that it can be computed off-line, finding the SVD can in any case be computationally intractable for vast databases. To solve this issue, various researchers have analyzed incremental techniques that changed current SVD without recomputing it from scratch [16,17]. Table 1 show the details of several proposals.

3 Dimensionality Reduction

We live in the age of BD where we do not have just a handful observations and variables; possibly often hundreds or even thousands of variables that we need to analyze, identify important trends, patterns, and to gain some insights about the businesses or for profit organizations to make policy decisions or even to do some basic research. Hence, we have many variables against which we have many observation stored in the same table. Now problem is how out of many observations select smaller group that contains chunk of observations. On the other hand we might be overwhelmed by the sheer number of variables in the data sets and some variables further more may be highly correlated or highly similar to each other creating additional problems with their interpretation and modeling itself. Hence, we might be interested to reduce the number of variables.

Second issue, we might be interested to revolves the way too many variables within our data sets and we're interested to see how our variable hang together, and how they can describe the datasets in the most efficient way. The variables may described very similar things and we're looking for the underlying similarity. Then group those variables together into a single broad dimensions that will describe our data set most efficiently. It is not advisable to enter all the variables in a single model because it's very often quite inefficient, computationally expensive, and their are high correlations among variables. PCA is especially helpful in this situation.

To reduce the dimensions of data apply cluster analysis over it. Further to reduce the dimensions of constructs, PCA and exploratory factor analysis give good results. In this paper, we have discussed the reduction of dimension of constructs or reduction in number of variables in existing data set. Next subsection present the PCA in detail.

3.1 Dimensionality Reduction Through Principle Component Analysis

PCA is a technique for extracting important factors (components) from a vast set of variables accessible in a dataset. It extricates low dimensional set of elements from a high dimensional dataset with an objective of getting as much information as possible. With a less factors, representation it turns out to be significantly more important. PCA is more valuable when managing three or more dimensional data. It is always performed on a symmetric correlation or covariance matrix. This implies that the matrix out to be numeric and have standardized data. First principal component is a linear combination of original predictor factors which catches the highest variance in the dataset. It decides the direction of most variability in the data. Higher the variability caught in first component implies more information caught by component. No other component can have variability higher than first principal component. The first principal component brings out to be a line which is nearest to the data i.e. it limits the sum of squared distance between a data point and the line. Likewise, we can also compute the second principal component. Second principal component is a linear combination of original predictors like first component which catches the rest of variance in the dataset and is uncorrelated with the first principal component outcome. That is, the correlation between first and second component should be zero. The direction of two components are orthogonal, if they are uncorrelated.

All succeeding principal component follows a similar idea, they catch the rest of variations without being correlated with the past component. The directions of these components are distinguished in an unsupervised way that means, the response variable is not used to decide the component direction. Thus, it is an unsupervised approach. As an example, M is a matrix, rows of which refers to the point in space, we can compute $M^T M$ and eigen pairs of that point. E, the matrix, which columns as the eigen vectors, ordered in such a way that largest eigenvalue comes first. Let the matrix L having the eigenvalues of $M^T M$ along the diagonal, in such a way that largest value comes first and 0's in other entries. Then, though $M^T Me = \lambda e = e\lambda$ for every eigen vector e and its related eigen value λ, it is understandable that:

$$M^T ME = EL \tag{1}$$

It has been observed that ME is the points of M changed into another coordinate space, in which, the first axis that is related to the largest eigen value, is critical. The variance of points along that axis is the most. The second axis, related to the second eigen pair, is the next noteworthy in the similar way, and this pattern proceeds for every eigen pairs. If it is desired that, M is transformed into a space having less dimensions, then the choice having the most important uses the eigen vectors related to the highest eigen values and discards the other eigen values, i.e., if E_k is the first k columns of E, then ME_k is the k-dimensional potrayal of M. Next subsection presents another DR technique, that is Singular Value Decomposition.

3.2 Dimensionality Reduction Through Singular Value Decomposition

SVD permits a accurate portrayal of any matrix, and furthermore SVD makes it simple to remove the less vital factors of that portrayal to deliver an approximate portrayal with any coveted number of dimensions. M is an $m \times n$ matrix The rank of M is r. Where the matrix rank r is the largest number of rows or columns that we can get for nonzero nonlinear combination of the rows which is the all-zero vector 0, in other words, a set of these rows or columns is independent of each other. Then,

- U be $m \times r$ column-orthonormal matrix. Each columns of this matrix is a unit vector and the dot product of any two columns is 0.
- V be $n \times r$ column-orthonormal matrix. V is utilized as its transposed form, so that the rows of V^T that are orthonormal.
- S be a diagonal matrix. Elements, that are not on the main diagonal are 0. S elements are known as the singular values of M.

If we take a very large matrix M by SVD components U, S, and V, however these three matrices are also extensive to store. Then,

$$M_{m\times n} = U_{m\times r} S_{r\times r} (V_{n\times r})^T \tag{2}$$

To diminish the dimensionality of the three matrices, the most ideal approach can be set the singular values that are smallest to zero. We can remove s columns of U and V, if the s smallest singular values are set to 0. Advantages of using SVD are:

- SVD gives best axis to project on, means, minimum sum of projection error.
- Minimum construction error.

But at the same time SVD have some gaps, which are:

- **Interpretability problem:** A singular vectors specifies a linear combination of all input columns and rows.
- **Lack of sparsity:** Singular vectors are dense.

4 Result and Discussion

In this section, we have compared PCA and SVD in terms of accuracy and mean square error. PCA works by finding the eigenvectors of the covariance matrix and ranking them by their respective eigenvalues. The eigenvectors with the greatest eigenvalues are the principal components of the data matrix. The matrix of eigenvectors in PCA are the same as the singular vectors from SVD, and the eigenvalues generated in PCA are just the squares of the singular values from SVD. While formally both solutions can be used to calculate the same principal components and their corresponding eigen/singular values, the extra step of calculating the covariance matrix in PCA can lead to numerical rounding errors when calculating the eigenvalues/vectors. Moreover, PCA gives the subspace that spans the deviations from the mean data sample as output, and SVD provides a subspace that spans the data samples themselves (or, a subspace that spans the deviations from zero).

4.1 Comparison of PCA and SVD in Terms of Accuracy

We have considered multivariate "Spam E-mail Dataset", of UCI Machine Learning Repository [18]. Before applying DR techniques accuracy was 93%. Here, our ultimate objective is to compare performance of PCA and SVD. Figure 2(a) show as number of attribute decreases, accuracy of PCA and SVD decreases. For some number of attributes, PCA gives maximum accuracy, but then drops drastically. But in SVD accuracy decreases gradually with decrease in attributes. It is important to note that PCA (5–7 min) takes lot of time compared to SVD (in seconds) to process around four thousands records. We have considered another dataset, "Wisconsin Breast Cancer Dataset" from UCI repository [18]. We have performed Same steps as performed in previous dataset, to analyse the performance of PCA and SVD. Initially the accuracy of data set was 97.54%. For this dataset, as number of attributes decreases, first accuracy of SVD increases, but then decreases gradually. But for PCA, accuracy decreases dramatically as number of attributes decreases, as shown in Fig. 2(b).

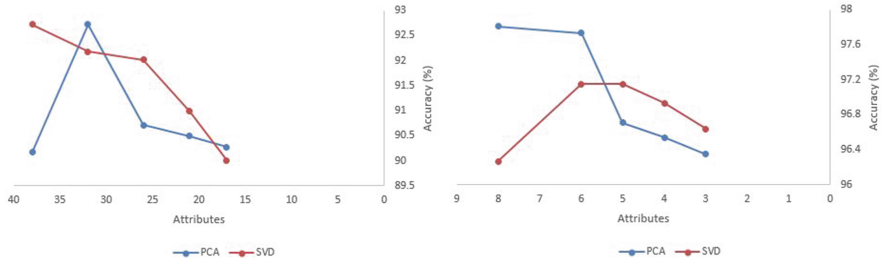

Fig. 2. (a) Spam E-mail dataset accuracy (%) Vs. no. of attributes, (b) Wisconsin breast cancer dataset accuracy (%) Vs. no. of attributes

4.2 Comparison of PCA and SVD in Terms of Mean Square Error

We have also compared PCA and SVD in terms of mean square error. For "Spam E-mail Dataset", mean square errors for PCA and SVD are almost same as shown in Fig. 3(a). For "Wisconsin Breast Cancer Databset" mean square errors for SVD is more than PCAb, as shown in Fig. 3(b).

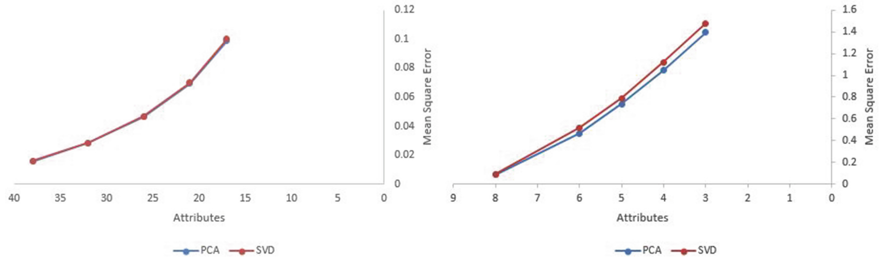

Fig. 3. (a) Spam E-mail dataset mean square error Vs. no. of attributes, (b) Wisconsin breast cancer dataset mean square error (%) Vs. no. of attributes

5 Conclusions

Their is an urgent requirement to process rapidly generated data with less storage space. Moreover this data is uncertain, redundant and inconsistent. Therefore, DR techniques comes in to picture, for fast processing of this data. Their are many approaches exist in the literature for DR, but we have discussed two of them, Principle Component Analysis and Singular Value Decomposition. We have compared the performance of both in terms of accuracy and mean square root. From comparison we have concluded that through SVD we get the "effective dimensionality" of a set of points. Moreover, for numerical reasons, it is preferred to use SVD. As it doesn't need to compute the covariance matrix which can introduce some numerical problems. Because there are some pathological cases where the covariance matrix is very hard to compute. So the SVD is numerically more efficient. Using the SVD to training data to diminish the dimension in an image gives good classification output. In future we will explore more DR approaches and apply tensor decomposition over these.

References

1. Gantz, J., Reinsel, D.: IDC, The Digital Universe (2014)
2. Swati, A., Ade, R.: Dimensionality reduction: an effective technique for feature selection. Int. J. Comput. Appl. **117**(3), 18–23 (2015)
3. Gupta, T.K., et al.: Dimensionality reduction techniques and its applications. J. Comput. Sci. Syst. Biol. **8**(3), 170 (2015)
4. Person, K.: On lines and planes of closest fit to system of points in space. Philos. Mag. **2**, 559–572 (1901)
5. Hotelling, H.: Analysis of a complex of statistical variables into principal components. J. Educ. Psychol. **24**(6), 417 (1933)
6. Jollie, I.T.: Principal Component Analysis. Springer, New York (1986)
7. Oja, E.: Simplifed neuron model as a principal component analyzer. J. Math. Biol. **15**(3), 267273 (1982)
8. Terence, D.: An optimality principle for unsupervised learning. In: NIPS, pp. 11–19 (1988)

9. Kung, S.Y., Diamantaras, K.I.: A neural network learning algorithm for adaptive principal component extraction (APEX). In: International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1990, pp. 861–864 (1990)
10. Rubner, J., Tavan, P.: A self-organizing network for principal-component analysis. EPL (Europhysics Letters) **10**(7), 693–696 (1989)
11. Baldi, P., Hornik, K.: Neural networks and principal component analysis: learning from examples without local minima. Neural Netw. **2**(1), 53–58 (1989)
12. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. JHU Press, Baltimore and London (2012)
13. Henry, E.R., Hofrichter, J.: Singular value decomposition: application to analysis of experimental data. Methods Enzymol. **210**, 129–192 (1992)
14. Deerwester, S., Harshman, R., et al.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–397 (1990)
15. Sarwar, B., et al.: Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document (2000)
16. Brand, M.: Fast online SVD revisions for lightweight recommender systems. In: Proceedings of the International Conference on Data Mining, pp. 37–46. SIAM (2003)
17. Sarwar, B., et al.: Incremental singular value decomposition algorithms for highly scalable recommender systems. In: Fifth International Conference on Computer and Information Science, pp. 27–28 (2002)
18. Lichman, M.: UCI Machine Learning Repository (2013)